WIRELESS LAN SECURITY (IEEE 802.11b)

A Thesis

Submitted to the Department of Computer Science and Engineering

of

BRAC University

By

Iftheker Mohammad

Student ID: 03201076

&

Mohammad Ashik Elahi

Student ID: 03201016

In Partial Fulfillment of the

Requirements for the Degree

of

Bachelor of Science in Computer Science

May 2008

DECLARATION

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of
Supervisor

Signature of
Authors

ACKNOWLEDGMENTS

Special thanks to Sadia Hamid Kazi, Senior Lecturer, BRAC University for guide us to complete this work and give us time out of her busy schedules to consider this work. We also acknowledge Risat Mahmud Pathan, Senior Lecturer, BRAC University as giving us the preliminary idea of wireless network.

ABSTRACT

There is much regulatory and standards work in the area of network security, especially in wireless network. The wireless LAN standard IEEE 802.11b provides a mechanism for authentication and encryption. This paper describes the security of Wireless Local Area Networks based on the IEEE 802.11 standard. Such networks are also known as Wi-Fi Networks or WLANs. We have analyzed different tools that used, to attack a wireless network. We have successfully detect different types of attack and found some flaws in WLAN. We conclude this research with several recommendations that will help improve security at a wireless deployment site.

TABLE OF CONTENTS

Page

CHAPTER I

INTRODUCTION

1.1 Overview of WLAN:

Wireless LAN [1] is a member of the IEEE 802 family of specifications for Local Area Networks which allows computers to get connected to a network through wireless. And because of wireless is a shared medium, everything that is transmitted or received over a wireless network can be intercepted.

Protocol	Release Date	Op. Frequency	Throughput (Typ)	Data Rate (Max)	Modulation Technique	Range (Radius Indoor) Depends, # and type of walls	Range (Radius Outdoor) Loss includes one wall
Legacy	1997	2.4 GHz	0.9 Mbit/s	2 Mbit/s		~20 Meters	~100 Meters
802.11a	1999	5 GHz	23 Mbit/s	54 Mbit/s	OFDM	~35 Meters	~120 Meters
802.11b	1999	2.4 GHz	4.3 Mbit/s	11 Mbit/s	DSSS	~38 Meters	~140 Meters
802.11g	2003	2.4 GHz	19 Mbit/s	54 Mbit/s	OFDM	~38 Meters	~140 Meters
802.11n	June 2009 (est.)	2.4 GHz 5 GHz	74 Mbit/s	248 Mbit/s		~70 Meters	~250 Meters
802.11y	June 2008 (est.)	3.7 GHz	23 Mbit/s	54 Mbit/s		~50 Meters	~5000 Meters

Table 1.1 Characteristics of different version of wireless network protocols.

A survey had found that about 70% of wireless network have no security at all. Among them IEEE 802.11b [2] is the most popular and widely used protocol in wireless arena. So, we choose 802.11b and it's really important to ensure security in WLAN.

1.2 How does WLAN work?

There are two basic modes of operation specified in the standard. The most commonly used mode is the infrastructure mode. The infrastructure mode allows for either one of the entities to be an access point.

In ad-hoc mode all entities are considered clients. Ad-hoc mode may also be referred to as independent mode. Stations in ad-hoc mode participate in an ad-

hoc network, likewise if they are in infrastructure mode they participate in an infrastructure network. Interface of a client or access point contains a radio and an antenna. To avoid interference and allow networks to operate in the same locations, IEEE 802.11 specifies groups of frequencies that may be utilized by a network.

CHAPTER II

Methodology

2.1 Working Methods:

Our Methodology of solving the security problems are as follows:

- Setup a Lab environment.(AP, Valid client, attacker.)
- Analyzing well-known attacking tools.
- Trace data sent and received by attacking tools.
- Analyze captured frame to detect attacks type.
- Give solution based on that recognized attacks.

We had configured and Access Point, one valid client and an attacker. We also successfully transferred and received a fixed amount of data between the AP and the client. Attacker use different tools to gain access to AP and meanwhile we trace the data coming in and going out from AP.

2.2 Analyzed Tools:

Tools we have used are: analyzed tools

"*Tehereal*" for tracing data sent by attacking tools
"*Wireshark*" to sniff network
"*NetStumbler*" to test WLAN sniffing
"*Kismet*" to test passive sniffing
"*Wellenreiter"* for WLAN MAC spoofing
"*AirSnrot*" for WEP cracking
"*AirJack*" to test Man-In-the-Middle attack

CHAPTER III

Analyzing Wi-Fi Network Traffic

3.1 Overview

Every packet transmitted in Wi-Fi networks contains bits of information used to maintain the various layers of the communication. Although packets may be encrypted in Wi-Fi networks, they still have plaintext headers. The headers are valuable to anyone analyzing the network. The entire MAC (Media Access Control) frame displayed in the above figure is easily available to user-space tools in Linux. All packets in a Wi-Fi network conform to the MAC frame format. The Frame Control field specifies which type of payload the MAC frame transports. There are three main types of packets and many subtypes.

- **Data Frames** – Carry Higher Level Protocol data.
- **Control Frames** – RTS, CTS, ACK etc.

RTS stands for Request to send. That means When Source want to send data to destination RTS frame is sent.

CTS stand for clear to send. When the destination wants to say the source that, it is ready to receive.

- **Management Frames** – Beacon, Probe, Association, Authentication, De-authentication etc.

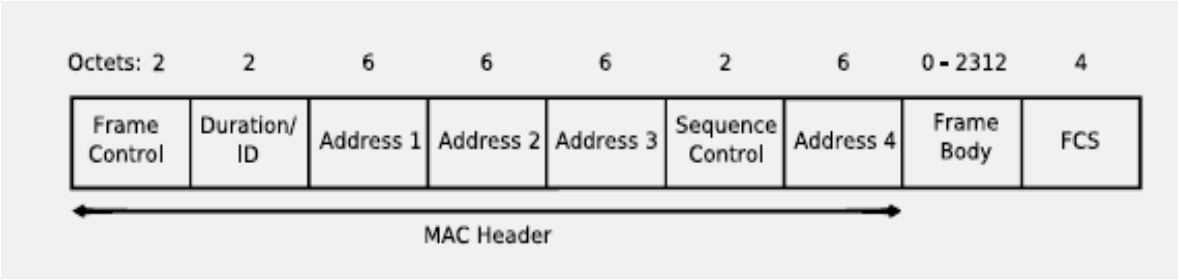

Fig. 3.1 Basic WLAN MAC Frame Format [3]

In this frame 2 bytes sequence number is important.

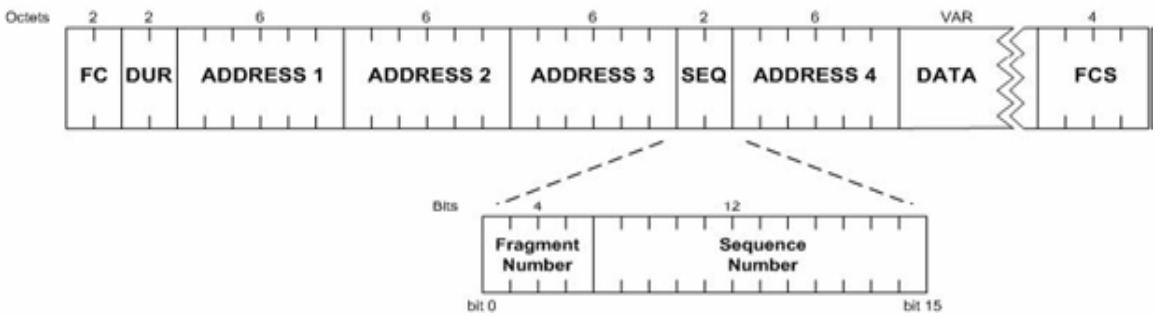

Fig. 3.2 Basic WLAN Frame sequence number

The IEEE dedicated two bytes of every 802.11 frame for sequence control fields: four bits for a fragment number and 12 bits for a sequence number. When management or data frames need to be fragmented they are transmitted in portions with a constant sequence number and incrementing fragment numbers for each portion of the packet. We will again focus on sequence number in FakeAP detection, Man-In-Middle attack detection.

3.2 Information about Data frame [3]

WEP or WPA encryption: B14 = 1

Type of payload: E.g. if the destination address is the broadcast address, and the size of the payload is 68 bytes, then it is very likely to be an Address Resolution Protocol (ARP) request.

Only data packets with Frame Capability:
ToDS = 1 and
FromDS = 1, are transmitted.

MAC address of access point:
In MAC header: Address 1, 2 or 3.

MAC address of mobile stations:
In MAC header: Address 1, 2, 3 or 4.

MAC address of wired stations:
In MAC header: Address 1, 2, 3 or 4.

3.3 Information about Control frames [3]

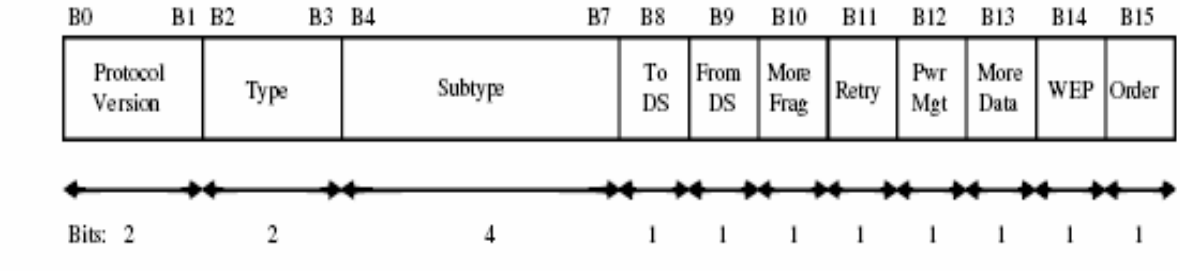

Fig. 3.3 Control frame field.

Network is part of a WDS:
ToDS = 1 and FromDS = 1.

Network is in ad-hoc mode:
ToDS = 0 and FromDS = 0; and Type = Data.

NeNetwork is in infrastructure mode:
ToDS = 1 or FromDS = 1; and Type = Data.

3.4 Information about Management frames [3]

Some management frames transmit many parameters about the network. The beacon frame is one of them. Access points will broadcast beacon frames to inform stations that they are available. The frames provide enough information for a client to be able to join the network. However management frames are strictly used to administer the network connections. They do not send any data from the application layer.

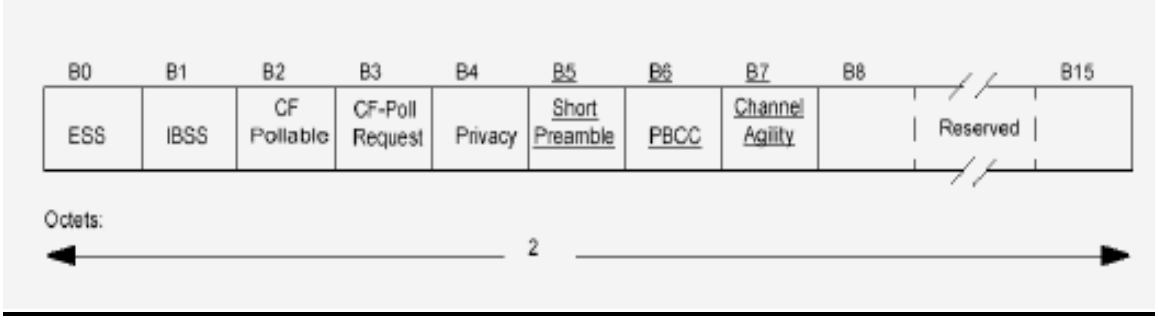

Fig. 3.4 Management frame field.

Network is in ad-hoc mode: B0 = 0 and B1 = 1.

WEP is required: B4 = 1.

Beacon interval: is the time between each transmitted beacon frame (typically100*1024μs = 100ms).

Service Set Identity (SSID) [4]**:** a string of maximum 32 bytes/ characters that gives a human readable identification of a Wi-Fi network. It also serves another purpose to group together multiple access points to form a network of collaborating access points (AP). Supported rates: a"list" of supported transmit rates in the network.

Extended supported rates: other supported rates.

Channel: the channel the network is operating on.

Network is in infrastructure mode: B0 = 1 and B1 = 0.

Fact	Frame	Requirements
WDS	Data	1 frame
Ad-hoc/Infrastructure	Beacon/Probe/Data	1 frame
Network range	Any	3 frames and GPS
Client/Access point location	Any	3 frames and GPS
WEP	Beacon/Probe/Data	1 frame
WPA	Beacon/Probe/Data	1 frame
SSID	Beacon/Probe	1 frame
Access point MAC address	Any	1 frame
Client MAC address	Probe Request/Data	1 frame
Wired client MAC address	Data	1 frame
Contents of data	Data	Intelligent guess

Fig. 3.5 Information available from an analysis of Wi-Fi frames Size

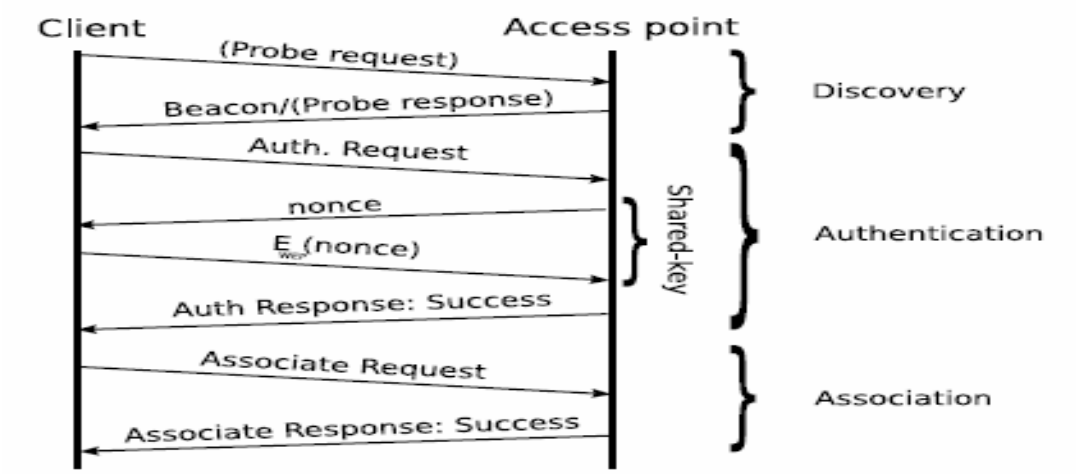

Fig. 3.6 this figure illustrates the basic protocols and flow of frames when connecting to an access point

First the client will detect access points either by sending a probe request and receiving a probe response, or purely by looking at the beacon frames frequently transmitted by an access point. Upon discovery, the client may try to authenticate to the access point. If successfully authenticated, the client may try to associate with the access point by sending an association request. If permitted by the access point the client will receive a positive association response. WEP is skipped here (open system authentication is used), and the real authentication is performed after association.

3.5 Frame formats:

In wireless network communication there are two types of frame available.

- **Long preamble:** Compatible with legacy IEEE* 802.11 systems operating at 1 and 2 Mbps (Megabits per second) PLCP with long preamble is transmitted at 1 Mbps regardless of transmit rate of data frames. Total Long Preamble transfer time is a constant at 192 Micro sec.

- **Short preamble:** is transmitted at 1 Mbps and header at 2 Mbps. Total short Preamble transfer time is a constant at 96 micro seconds**.** Short preamble more efficient than long preamble.

The preamble is used to communicate to the receiver that the data is on its way. Technically speaking, it is the first portion of the Physical Layer Convergence Protocol (PLCP). Each preamble have atmost 2312 bytes of payload. The payload allows the receiver to acquire the wireless signal and synchronize itself with the transmitter.

CHAPTER IV

WLAN Security Issue

4.1 Overview:

The pictures below shows you a small scenario, where the attacker can take position. As it is a wireless network, an attacker can took position in anywhere the network available. It can be an individual try to gain access to AP, or may be a fake AP try to be the part of a network.

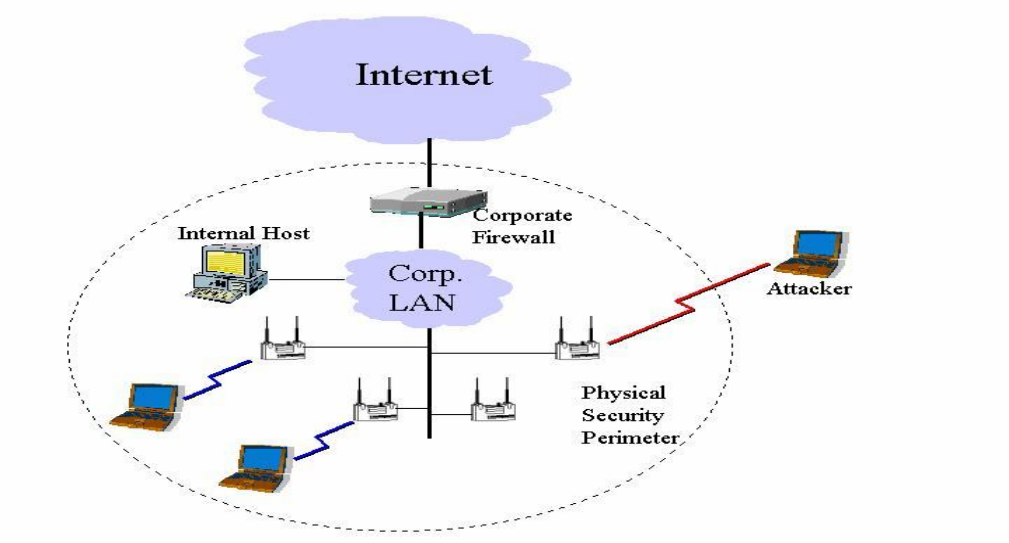

Fig. 4.1 Parking Lot attacker in WLAN

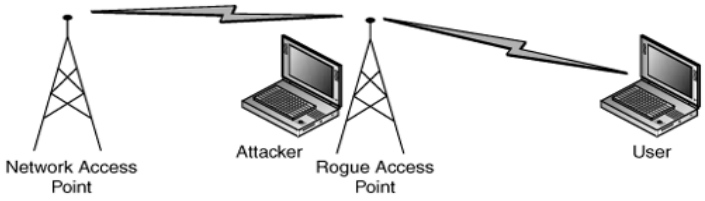

Fig. 4.2 a rouge Access point

4.2 Wireless LAN sniffing:

Sniffing is much easier to do in wireless network. One doesn't needs to be physically connected to the network to sniff. Generally sniffing is done by a sniffing program which copies of a network packet sent by machine A intended to be received by machine B.

There are two types of sniffing tools; one is active sniffer and other one passive sniffer. Active sniffer uses probe request frames on each channel where it is able to detect wireless activity (to avoid blocking the discovery process on frequencies that are not in use). When an AP comes within range of the client and receives a probe request frame it will typically respond with a probe response frame containing the network ESSID. Passive sniffer utilizes a completely passive method of WLAN discovery known as radio frequency monitoring (RFMON). A client with a wireless card that is configured in RFMON mode will be able to capture all RF signals on the channels to which it is configured to listen. While in RFMON mode, wireless clients are unable to transmit any frames; their cards are only able to receive, and therefore capture traffic. This limits the client to reporting only current or recorded network traffic.

4.2.1 NetSstumbler:

A good example of active sniffer is NetSstumbler. NetStumber is a very popular tool for Windows users, utilizing active scanning through the use of probe requests sent to a broadcast address with a broadcast BSSID and an unspecified ESSID (length of 0). The probe requests are difficult to identify definitively as NetStumbler activity since NetStumbler utilizes the active scanning method described in the IEEE 802.11 specifications without anomalous characteristics.

4.2.2 Kismat:

Kismet [5] detects the presence of wireless networks, including those with hidden SSIDs. It can discover and report the IP range used for a particular wireless network, as well as its signal and noise levels. Kismet can also capture or "sniff" all network management data packets for an available wireless network. Kismet works by putting the wireless client adapter into RF monitor mode. While in so-called "rfmon" mode, the wireless client is not (and cannot be) associated with any access point. Instead, it listens to all wireless traffic. Consequently, your wireless card cannot maintain a functional network connection while under Kismet control.

4.3 WLAN MAC address spoofing :

The phrase “MAC address spoofing” [6] in this context relates to an attacker altering the Manufacturer assigned MAC address to any other value. Nearly all 802.11 cards in use permit their MAC addresses to be altered, often with full support and drivers from the manufacturer. Using Linux open-source drivers, a user can change their MAC address with the ifconfig tool. Windows users are commonly permitted to change their MAC address by selecting the properties of their network card drivers in the network control panel applet.

An attacker may choose to alter their MAC address for several reasons, including obfuscating their presence on a network, to bypass access control lists, or to impersonate an already authenticated user. Each is explained in greater detail as follows:

4.3.1 Obfuscating network presence:

An attacker might choose to change their MAC address in an attempt to evade network intrusion detection systems (NIDS). A common example is an attacker executing a brute- force attack script with a random MAC address for each successive connection attempt.

4.3.2 Bypassing access control lists:

Used as a basic form of access control on WLANs, administrators typically have the option to configure access points or neighboring routers to permit only registered MAC addresses to communicate on the network. An attacker could circumvent this form of access control by passively monitoring the network and generate a list of MAC addresses that are authorized to communicate. With the list of authorized MAC addresses in hand, an attacker is free to set their MAC address to any of the authorized addresses, bypassing the intended security mechanism.

4.3.3 Authenticated user impersonation:

Certain hardware WLAN security authentication devices rely on matching user authentication credentials to the source MAC address of a client. After a user has successfully authenticated, the security gateway permits traffic based on a dynamic list of authorized MAC addresses. An attacker wishing to avoid the security of the device, s/he only needs to monitor the network activity for an authorized client’s MAC address. Then alter their MAC address to match the authenticated client before communicating on the network.

4.4 Fake Access Point:

FakeAP [7] is a Perl script that utilizes the HostAP client drivers and a dictionary word list to generate 802.11 beacon frames in an attempt to fool NetStumbler, Kismet and other wireless LAN discovery applications. In order to make the generated traffic appear valid, FakeAP supports changing transmit signal strength and MAC addresses for each unique SSID advertisement. The authors of FakeAP were resourceful in their MAC address generation code, using a list of allocated OUIs [8] with three trailing random octets to generate a MAC address that will escape the anomalous MAC prefix detection method described above.

4.5 Wired Equivalent Privacy (WEP):

WEP [9] is a shared-secret key encryption system used to encrypt packets transmitted between a station and an AP. WEP incorporates two main types of protection: a secret key and encryption. The secret key is a simple 5- or 13-character password that is shared between the access point and all wireless network users. This key is all-important to WEP in that it is also used in the encryption process to uniquely scramble each packet of information with a unique password. This ensures that if a hacker cracks one packets key, he won't be able to view every packet's information.

To do this, WEP defines a method to create a unique secret key for each packet using the 5- or 13-characters of the pre-shared key and three more psuedo-randomly selected characters picked by the wireless hardware. For example, let's assume that our pre-shared key was "games". This word would then be merged with "abc" to create a secret key of "abcgames", which would be used to encrypt the packet. The next packet would still use "games", but concatenate it this time with "xyz" to create a new secret key of "xyzgames". This process would randomly continue during the transmission of data. This changing part of the secret key is called the Initialization Vector because it initializes the encryption process for each packet of data sent.There exists some WEP cracking software to crack WEP. Normally it will take hours to check and crack WEP secret key depending on how many bits it used to encrypt. An example of a WEP cracking tool is AirSnort.

It is important to understand the basics of XOR when discussing RC4 and WEP because it is used in the encryption process to create the encrypted data. XOR is just a simple binary comparison between two bytes that produces another byte as a result of a simple process. In short, it takes each corresponding bit in a byte and compares them by asking "Is this bit different from that bit?" If the answer is yes, the result is 1; otherwise it is a 0. Figure illustrates below.

Original bit	XOR bit	Resulting bit
1	**1**	**0**
0	**0**	**0**
1	**0**	**1**
0	**1**	**1**

Table 4.1 XOR Byte Comparison

4.5.1 AirSnort:

The most commonly used tool for WEP key extraction is the Linux program AirSnort. An intruder using AirSnort would surreptitiously collect wireless network traffic of the target network. When enough frames have been collected from the network, AirSnort can determine the WEP key of the network by examining the “weak” frames. It usually takes only a few hours to collect enough frames. Manufacturers have released updated firmware that addresses the transmission of such weak frames; however, a network remains vulnerable if a client continues to use an outdated wireless network adapter.

4.6 Man in the middle attack:

Assume that station B was authenticated with C, a legitimate AP. Attacker X is a laptop with two wireless cards. Through one card, he will present X as an AP. Attacker X sends De-authentication frames to B using the C’s MAC address as the source, and the BSSID he has collected. B gets de-authenticated and begins a scan for an AP and may find X on a channel different from C. There is a race condition between X and C. If B associates with X, the MITM attack succeeded. X will re-transmit the frames it receives from B to C, and the frames it receives from C to B after suitable modifications.

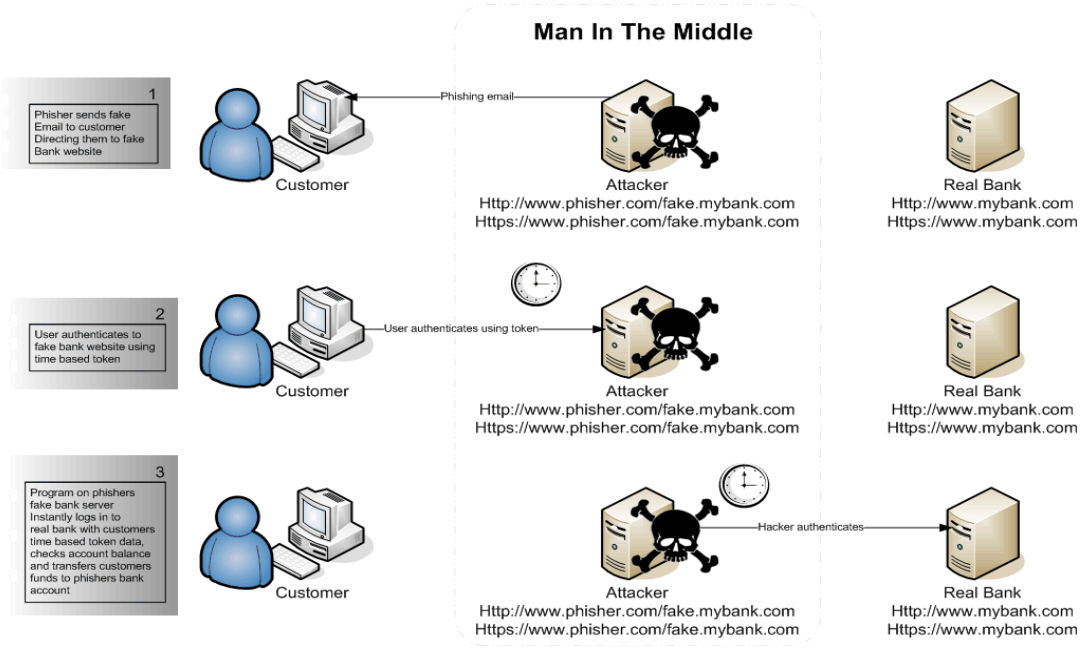

Fig. 4.3 Man in the middle attack.

The package of tools called AirJack:

Includes a program called monkey_jack that automates the MITM attack. This is programmed well so that the odds of it winning in the race condition mentioned above are improved. Attacker on machine X inserts himself between two hosts B and C by (i) poisoning B so that ARP poisoning:

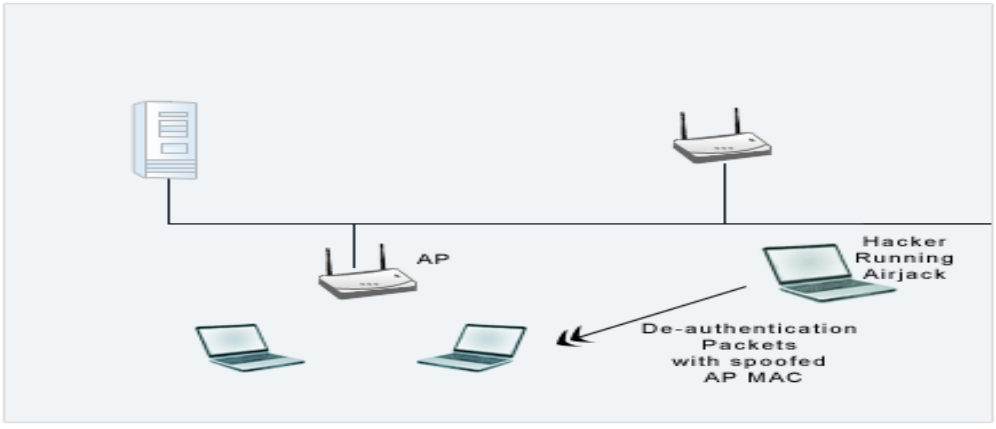

Fig. 4.4. Denial of service using AirJack

4.6.1 AirJack:

AirJack is a suite of tools that is designed as a proof-of-concept to establish layer 1 man- in-the-middle attacks against 802.11 networks. Included in this toolset is “wlan-jack”, a tool to perform a denial-of-service attack against users on a target wireless network; it works by sending spoofed de-authenticate frames to a broadcast address, purportedly from the network access point’s MAC address.

Using the wlan-jack tool is simple; an attacker only has to be in range of the clients on the wireless network to be effective. This tool works because the clients that receive the de-authenticate frames believe they have been sent from the access point they are currently associated with. Since wlan-jack has to spoof the MAC address of the target access point, we have the opportunity to identify this traffic as anomalous based on sequence number analysis.

CHAPTER V

Security Flaws Detectaction

5.1 NetStumbler detection

In a posting to the Kismet Wireless mailing list on March 28 2002, Mike Craik first identified a unique pattern that can be used to identify NetStumbler traffic. LLC-encapsulated frames generated by NetStumbler will use an organizationally unique identifier (OUI) of 0x00601d and protocol identifier (PID) of 0x0001. NetStumber also uses a data payload size of 58 bytes containing a unique string that can be used to identify the version of NetStumbler.

NetStumble	Payload String
3.2.0	Flurble gronk bloopit, bnip Frundletrune
3.2.3	All your 802.11b are belong to us
3.3.0	intentionally blank 1

Table 5.1 Payload string of different version of netstumbler

It is possible to detect NetStumbler by analyzing the probe requests coming from clients with above like format. So,

"If probe request frame = (wlan.fc.type_subtype = 0x08 and llc.oui = 0x00601d and llc.pid = 0x0001) and (data[14:4] = 69:6e:74:65 and data[18:4] = 6E:74:69:6f and data[22:4] = 6e:61:6c:6c and data[26:4] = 79:20:62:6c and data[30:4] = 61:6e:6b:20) then NetStumbler detected".

The following detect is taken from NetStumbler version 3.2.3 using a Windows 2000 host for discovery using ethereal [10].

IEEE 802.11
Type/Subtype: Data (32)
Frame Control: 0x0908
Version: 0

Type: Data frame (2)
Subtype: 0
Flags: 0x9

DS status: Frame is entering DS (To DS: 1 From DS: 0) (0x01)
.... .0.. = Fragments: No fragments
.... 1... = Retry: Frame is being retransmitted
...0 .... = PWR MGT: STA will stay up
..0. .... = More Data: No data buffered
.0.. .... = WEP flag: WEP is disabled
0... .... = Order flag: Not strictly ordered

Duration: 258
BSS Id: 00:50:18:07:13:92 (ADVANCED_07:13:92)
Source address: 00:02:2d:52:cb:27 (Agere_52:cb:27)
Destination address: 00:50:18:07:13:92 (ADVANCED_07:13:92)
Fragment number: 0
Sequence number: 3057

Logical-Link Control
DSAP: SNAP (0xaa)
IG Bit: Individual
SSAP: SNAP (0xaa)
CR Bit: Command

Control field: U, func = UI (0x03)
000. 00.. = Unnumbered Information
.... ..11 = Unnumbered frame

Organization Code: Unknown (0x00601d)
Protocol ID: 0x0001
Data (58 bytes)

0000 00 00 00 00 41 6c 6c 20 79 6f 75 72 20 38 30 32 **....All your 802**
0010 2e 31 31 62 20 61 72 65 20 62 65 6c 6f 6e 67 20 **.11b are belong**
0020 74 6f 20 75 73 2e 20 20 20 20 20 20 fe ca ba ab **to us.** *....*
0030 ad de 0f d0 4f 45 43 45 46 46 **....OECEFF**

5.2 Wellenreiter detection:

There are several tools to do the MAC spoofing attack. Among them Wellenreiter [11] is most useful and popular among Linux users. Wellenreiter is a Perl/Gtk+ application for use on Linux systems. Wellenreiter only supports network discovery through RFMON packet capture, but includes some additional features using a second 802.11 network card, including ESSID brute- forcing and automatic network association. It is through these features that we are able to identify Wellenreiter-generated traffic.

When an ESSID brute-force attack is initiated, Wellenreiter will use the Linux "iwconfig" program to temporarily set its ESSID to "this_is_used_for_wellenreiter". The MAC address will also be set to a random values between “0x00” and “0xFF” and then prepends a leading value of “0x00” to avoid generating MAC addresses that conflict with reserved and multicast space. A trace of Wellenreiter v1.6 traffic attempting to brute-force the ESSID of an access point appears as follows:

scornik:~ $ tethereal –r wellenreiter.dmp –n –R “wlan.fc eq 0x0040”
167 16.117154 00:03:b7:47:e2:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
168 16.133680 00:03:b7:47:e2:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
171 16.239248 00:84:b9:5a:ce:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
177 16.504359 00:84:b9:5a:ce:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
178 16.538390 00:0e:ef:e0:3d:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
183 16.885264 00:0e:ef:e0:3d:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
185 16.918787 00:a6:8b:cd:f1:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
191 17.270278 00:a6:8b:cd:f1:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
192 17.303924 00:8a:7c:29:56:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
198 17.653241 00:8a:7c:29:56:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
199 17.674149 00:df:f7:a1:8a:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
206 18.041326 00:df:f7:a1:8a:40 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
scornik:~ $

In this example every second probe request frame appears to be coming from a different MAC address, all destined to the broadcast address. If we compare the

MAC address prefixes to the allocated OUI listing by the IEEE we discover that only one of the five MAC prefixes is listed as being allocated by the IEEE.

With this information, we can detect anomalous MAC addresses based on their prefix information. With a current list of allocated OUIs from the IEEE, we can monitor network activity and identify the OUIs for MAC addresses that are not presently allocated.

Unfortunately, a clever attacker can easily circumvent this detection technique. Since the list of allocated OUIs is publicly available, an attacker could modify their randomization procedure to build a hash of allocated prefixes and randomly select an allocated OUI, and then append a random 24-bit value for the remaining three octets of the MAC address. This technique was first demonstrated in the FakeAP tool that generates multiple SSID beacon frames with varying MAC addresses, power output, and SSID values in an effort to fool network discovery applications into thinking they have discovered a plethora of wireless networks. Fortunately, we can detect the presence of FakeAP and other, more devious WLAN attack tools through WLAN sequence number analysis.

5.3 FakeAP detection

We can detect FakeAP activity by monitoring the sequence number value for sequential increments with changing source MAC address, BSSID and SSID values. In an environment with a dense deployment of wireless access points, we would typically see multiple beacon frames from varying source addresses containing different SSID names; using FakeAP, we will see similar activity as it cycles through its dictionary list of SSID values and random MAC address generation procedure. The firmware of the Prism wireless NIC will be indifferent to changing wireless parameters, however, maintaining a sequential order to the sequence number value.

The following packet capture was generated in a lab environment using FakeAP v0.3.1, with the output from tethereal trimmed for brevity:

scornik:~ $ tethereal –r fakeap.dmp –n –R “wlan.fc eq 0x0080”

IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:02:b3:cb:28:64 (00:02:b3:cb:28:64)
BSS Id: 00:02:b3:cb:28:64 (00:02:b3:cb:28:64)
Fragment number: 0
Sequence number: 2055
IEEE 802.11 wireless LAN management frame

Tagged parameters (22 bytes)
Tag Number: 0 (SSID parameter set)
Tag interpretation: macro

IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:02:a5:a5:75:a5 (00:02:a5:a5:75:a5)
BSS Id: 00:02:a5:a5:75:a5 (00:02:a5:a5:75:a5)
Fragment number: 0
Sequence number: 2056
IEEE 802.11 wireless LAN management frame
Tagged parameters (26 bytes)
Tag Number: 0 (SSID parameter set)
Tag interpretation: breathing
IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:02:a5:a5:75:a5 (00:02:a5:a5:75:a5)
BSS Id: 00:02:a5:a5:75:a5 (00:02:a5:a5:75:a5)
Fragment number: 0
Sequence number: 2057
IEEE 802.11 wireless LAN management frame
Tagged parameters (26 bytes)
Tag Number: 0 (SSID parameter set)
Tag interpretation: breathing

IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:02:2d:7d:c5:5f (00:02:2d:7d:c5:5f)
BSS Id: 00:02:2d:7d:c5:5f (00:02:2d:7d:c5:5f)
Fragment number: 0
Sequence number: 2058
IEEE 802.11 wireless LAN management frame
Tagged parameters (29 bytes)
Tag Number: 0 (SSID parameter set)
Tag interpretation: intercession
scornik:~ $

Here we can see that, we see unique source MAC addresses, BSSIDs and SSIDs that would normally appear as traffic generated by multiple WLAN access points. If we examine the sequence number parameter, however, we see a sequential pattern that indicates that this traffic is being generated by a single host and not multiple access points.

5.4 AirSnort detection:

Examining AirSnort's source code revealed that it uses a simple algorithm to determine if a frame is weak. Weak frames are then added to a pool to be analyzed later by another algorithm to determine the WEP key.

The weak frame detection algorithm relies on simple pattern matching of the initialization vector (IV) of the frame. The IV itself consists of three bytes. AirSnort considers a frame weak if the first byte of the IV has a value between 2 and 16 inclusive and the second byte has a value of 255. Each three-byte pattern is only added once to the pool.

Creating decoys for AirSnort is a simple task. The decoy frames contain mostly valid data except that the IV is a randomly generated number that matches the criteria described above. Also, at least one byte of "encrypted" data should be randomly generated. AirSnort requires only a few such frames and would cease capturing frames when the key has been extracted. Our implementation used sequential numbers for the third byte while the first byte and encrypted data are a single random byte each.

Other required fields are also filled in. First, the frame is marked as an encrypted data frame. The BSSID is the actual one configured by the user and the source address is also set from this value. The destination is randomized since it is ignored by AirSnort. To prevent degradation of the network, the decoy frames can only be sent intermittently at a user-controlled rate. Given that the system only has to protect the valid network from intrusion, the decoys are broadcasted only on the valid channel.

5.5 Airjack detection:

In order to detect anomalies in sequence numbers, we need to first establish a pattern of legitimate sequence number activity for each MAC address we wish to monitor. In the following detect, the sequence numbers for the legitimate source MAC "00:e0:63:82:19:c6" are in the range 2966 – 2971. Knowing this pattern, we can easily identify the deauthenticate frames as being illegitimate, although they purport to have originated from the source MAC "00:e0:63:82:19:c6". This packet capture was generated in a lab environment using AirJack [12] v0.6.2-alpha.

iftheker:~ $ tethereal –r airjack.dmp –n –R "wlan.sa eq 00:e0:63:82:19:c6"
IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)

BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 2966
IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 2967
IEEE 802.11
Type/Subtype: Probe Response (5)
Destination address: 00:60:1d:f0:91:56 (00:60:1d:f0:91:56)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 2968
IEEE 802.11
Type/Subtype: Deauthentication (12)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 1335
IEEE 802.11 wireless LAN management frame
Fixed parameters (2 bytes)
Reason code: Previous authentication no longer valid (0x0002)
IEEE 802.11
Type/Subtype: Deauthentication (12)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 1336
IEEE 802.11 wireless LAN management frame
Fixed parameters (2 bytes)
Reason code: Previous authentication no longer valid (0x0002)
IEEE 802.11
Type/Subtype: Beacon frame (8)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 2969
IEEE 802.11

Type/Subtype: Probe Response (5)
Destination address: 00:60:1d:f0:91:56 (00:60:1d:f0:91:56)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 2970
IEEE 802.11
Type/Subtype: Deauthentication (12)
Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 1339
IEEE 802.11 wireless LAN management frame
Fixed parameters (2 bytes)
Reason code: Previous authentication no longer valid (0x0002)
IEEE 802.11
Type/Subtype: Probe Response (5)
Destination address: 00:60:1d:f0:91:56 (00:60:1d:f0:91:56)
Source address: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
BSS Id: 00:e0:63:82:19:c6 (00:e0:63:82:19:c6)
Fragment number: 0
Sequence number: 2971
iftheker:~ $

We can quickly identify the frames with sequence numbers in the range of 1335-1339 as illegitimate management traffic.

CHAPTER VI

Conclusion

6.1 Conclusion:

Based upon the results from our research efforts and proposed solutions, we feel that it's possible to implement a complete WIDS (Wireless Intrusion Detection System) in the application layer based on fingerprinting known attacking tool's attack pattern. It's almost impossible and not feasible enough to deploy hardware intrusion detection system. Thus our future plan is to analyze more attacking tools, find their pattern and implement software based complete intrusion detection system.

REFERENCES

[1] Jim Geir, *Wireless LANs: Implementing Interoperable Networks*, MacMillan Technical Publishing, USA, pp. 25-26, 138-142.

[2] Harold Davis & Richard Mansfield: *The Wi-Fi Experience: Everyone's Guide to 802.11b Wireless Networking*. ISBN: 81-7635-671-9

[3] *The Handbook of Information Security*, Hossein Bidgoli (Editor-in-Chief), John Wiley & Sons, Inc., 2005

[4] Wi-Fi Security – *by Hallvar Helleseth, Department of Informatics, University of Bergen, June 2006(Thesis for the master of science)*.

[5] Detection of Kismet www.kismetwireless.net

[6] Joshua Wright, *GCIH, CCNA*
Joshua.Wright@jwu.edu
http://home.jwu.edu/jwright/
www.willhackforsushi.com/papers/l2-wlan-ids.pdf
www.uninett.no/wlan/download/wlan-mac-spoof.pdf

[7] FakeAP. "*Black Alchemy Weapons Lab*." URL:
http://www.blackalchemy.to/project/fakeap/

[8] IEEE. "*IEEE OUI and Company_id Assignments*." URL:
http://standards.ieee.org/regauth/oui/oui.txt

[9] Attacks on RC4 and WEP; *Fluhrer, Mantin and Shamir ; Cryptobytes 2002.*

[10] Ethereal home page - great documentation and source for disassembling all sorts of network protocols - http://www.ethereal.com/

[11] Wellenreiter. "*Wireless LAN Discovery and Auditing Tool*" URL:
http://www.remote-exploit.org/

[12] AirJack. "*Advanced 802.11 Attack Tools." URL*:
http://802.11ninja.net/airjack/

[NOTES]